Financial Simulations on a Massively Parallel Connection Machine

James M. Hutchinson [1]
Stavros A. Zenios [2]

Report 90–04–01
HERMES Laboratory for Financial Modeling and Simulation [3]
Decision Sciences Department
The Wharton School
University of Pennsylvania
Philadelphia, PA 19104, USA

April, 1990.
Revised, September 1990.

FA90-1

[1] Thinking Machines Corporation, 245 First St., Cambridge, MA 02142.
[2] Decision Sciences Department, The Wharton School, University of Pennsylvania, Philadelphia, PA 19104, and Thinking Machines Corporation, Cambridge, MA.
[3] The HERMES Laboratory was established by a grant from Digital Equipment Corporation.

Contents

Abstract

The valuation of interest rate contingent claims — like mortgage-backed securities, corporate bonds with call options, single premium deferred annuities, collaterized mortgage obligations and so on — usually relies on simulation methods. Such methods of analysis integrate models of the term structure of interest rates with estimations of the contingent cash flows. These simulations are quite complex and computationally intensive.

In this paper we describe the development of massively parallel procedures for financial modeling simulations on systems like the Connection Machine. Several primitives — generic among multiple applications — are first presented, and the discussion culminates in an *option adjusted analysis* model for mortgage-backed securities. Implimentation of the primitives on a Connection Machine CM–2a with 4096 processors achieve sustained computing rates in the range 30 — 120 MFLOPS. The response time for a complete option adjusted analysis (1 – 2 seconds) makes the model usable for real time analysis and trading applications.

1 Introduction

It is difficult to find an area of the fixed income market that does not rely on analytic techniques for instrument pricing. Instruments like government bonds generate a fixed stream of cashflow payments to investors and their pricing is rather easy. For more complex instruments, however, the cashflow payments may depend on the prevailing interest rates. In these cases the pricing methodology has to account for plausible scenarios of the evolution of interest rates. Analysts often resort to statistical simulations to capture the effect of interest rate variations on the cashflow of these complex instruments. Interest rate sensitive instruments appear in several forms: European and American style options, callable and putable bonds, mortgage-backed securities and their derivatives, and products from the insurance industry such as guaranteed insurance products and single premium deferred annuities.

In this study we describe the development of massively parallel procedures for financial modeling simulations. We concentrate on a particular class of interest rate contingencies, *mortgage-backed securities* — MBS for short — which we describe in detail next. Nevertheless, the parallel computing techniques developed here can be used in a much broader range of applications.

Mortgage-backed securities are created when mortgages are pooled together, and investors purchase interest in the pool and receive pro-rated shares of the pool's cashflows. The issuing institution, or the mortgage originators, handle the transfer of funds and retain a service fee. These instruments facilitate the flow of funds from the ultimate lenders in the capital markets to the mortgage borrower. Lending institutions can escape the mismatch of a short term cost of funds (deposits) and the long term return from a portfolio of mortgage loans by converting their portfolio of illiquid long-term loans into holdings of liquid MBS, while reaping additional income from servicing fees. For other investors, MBS are an interesting alternative instrument given their potential for high return with little credit risk.

Mortgage-backed securities have emerged in the 1980's as an important class of securities. At second quarter of 1988 outstanding mortgage debt in the United States was approximately \$3.5 trillion, of which nearly 70% was in residential mortgages. This sum dwarfs the more established corporate and government debt markets. To date only some 25% of the outstanding residential debt has been securitized via the issuance of mortgage-backed securities, but MBS represent the fastest growing segment of the debt markets.

Growth in outstanding MBS was dramatic into the mid 1980's — from \$3 billion outstanding in 1979 to over \$900 billion by the end of 1988. Trading in MBS after issuance has also increased significantly in the last few years — from \$243 billion in 1981 to \$1.2 trillion in 1985. Interest in MBS is not restricted to the US alone. Almost 90% of all residential debt in Denmark has been securitized. In Canada, where the first issue took place as recently as January of 1987, there are today over 70 issues backed by mortgage loans in excess of \$500 million.

Mortgage-backed securities are complex and difficult to value as they embody features of both bonds and options. The homeowner's ability to prepay outstanding principal represents a *call* option on the underlying mortgage. For any specific mortgage within a pool whether this call option will be exercised, and if so when, is uncertain. Many factors outside the characteristics of the pool may affect the option's value. Some of these factors include the level, structure and history of interest rates, the market perception of future interest rates, total and disposable consumer income, and others. Adding to the complexity of early MBS has been the constant stream of innovative new derivative securities whose risk and return characteristics can bear little resemblance to the original MBS.

Several references discuss the emergence of mortgage-backed securities and their characteristics. For general interest we cite Pavel [1986], Pinkus et al. [1987] and Fabozzi [1987,1988], or the book by Platt [1986]. Pavel discusses securitization, the process by which the assets (loans or other receivables) of financial intermediaries are converted into securities. The economic rationale for securitization is presented. Pinkus et al. present a comprehensive introduction to the primary and secondary mortgage markets, discuss various mortgage types and the variety of securities that are available to investors. They review the problems inherent to analyzing the prepayment of principal and outline several approaches to MBS valuation. Fabozzi's volumes are compilations of published articles and research findings by some of the leading participants in the MBS market. A discussion of the Canadian MBS market and related valuation techniques are developed in Boyle [1989].

Understanding MBS is a complex process that relates the possible future paths of interest rates to the cashflows generated by the security. Such cashflows should take into account both payment of principal and interest, as well as prepayment of the mortgage (i.e., exercise of the underlying call option). The analysis is carried out using simulations to generate paths of interest rates, usually in monthly intervals for a period of 30 years. The state space from which simulations are drawn could be enormous. Consider, for example, a binomial lattice model of interest rates — as explained later in Section 2.1.2 — for this 30-year period. Short term interest rates can be at any one of 360 possible states at the end of the period, and the number of interest rate scenarios that lead to these states is 2^{360}. Even with the use of variance reduction techniques, a sample of several hundred paths — typically around 1000 — has to be analyzed in order to obtain pricing estimates within acceptable error bounds.

These simulations are quite complex and time consuming. Analyzing a single security takes several minutes of computer time on a large mainframe. Viewed in the context of portfolio management, when several securities have to be analyzed for their relative attractiveness, the task may indeed become formidable.

The computational complexities of these models led users to investigate super- and parallel computers as suitable platforms for such applications. The rapidly changing environment where the analyses are performed makes the quest for faster response times even more pressing. The interest of Wall Street analysts in advanced computing technology has been the topic of recent articles in popular magazines [4]. While several firms use time on a supercomputer to carry their simulations, no significant progress has been made in utilizing parallel computing technology. For example, Wallace and Wang [1989] and Foote et al. [1988] implemented their models for mortgage pricing on vector supercomputers — the CRAY X-MP and IBM 3090 respectively. However, we are not aware of any publications that discuss the use of parallel computer architectures in financial applications.

In this paper we report on the use of massively parallel architectures — like the Connection Machine CM–2 — for the simulation of interest-rate-contingent and path-dependent cash flows. Simulating multiple paths of interest rates is indeed an *embarrassingly parallel* procedure. Exploiting parallelism in performing the path-dependent calculations appears, on first examination, to be impossible. However, with suitable reformulation of the models and use of the parallel prefix primitives of the Connection Machine we are able to exploit parallelism both in executing multiple simulations and in performing the path-dependent calculations along each path. The result is a library of primitives that run efficiently on the Connection Machine and achieve sustained computing rates of 30 — 120 MFLOPS on a

[4] See the article "Supercomputers: Era Dawns on Wall Street" in the November 1989 issue of *Wall Street Computer Review* and the special report "Supercomputers Break Through" in the May 1988 issue of *Datamation*.

CM–2a with 4096 processors. The library is used in building an *option adjusted analysis* model for mortgage-backed securities. This system can price mortgage-backed securities in real time. It is, therefore, now feasible not only to price complex instruments in real time, but also to compare alternative instruments under a host of interest rate scenarios and other exogenous factors that affect their price. Analysing, for example, a subset of the large variety of pools of mortgage pass through securities that exist today in the U.S. (say 3000) would require 10 hours on a CRAY X-MP supercomputer and more than two weeks on an Apollo DN4000 workstation. This analysis can be carried out on a CM–2a with 8K processing elements in approximately 1 hour, and in less than 5 minutes on a fully configured CM–2 with 64K processing elements.

In Section 2 we discuss the methodology for valuing interest rate contingencies. Section 3 provides a brief overview of the Connection Machine CM–2. Section 4 provides details on the parallelization of key modules of the valuation methodology, and Section 5 reports timing results and comparisons with an identical model running on a CRAY X-MP vector supercomputer, a range of workstations and a mainframe. Concluding remarks are the topic of Section 6.

2 Valuation of Interest Rate Contingent Cashflows

In this section we review the methodology commonly adopted in the valuation of interest rate contingent cash flows as given, for example, in the generalized pricing framework of Jacob, Lord and Tilley [1987]. Applications in the mortgage-backed securities market are reported in Brazil [1988], Hayre and Lauterbach [1988] or Pinkus et al. [1987] and the valuation of single premium deferred annuities is discussed in Asay et al. [1989].

The general framework of the valuation analysis has three phases:

Phase I: Generate arbitrage free interest rate scenarios that are consistent with the prevailing term structure of interest rates.

Phase II: Generate cashflows along each interest rate scenario.

Phase III: Use the cashflows and the short term interest rates along each path to compute expected net present value of the cashflows, or to compute an *option adjusted spread* over the Treasury yield curve.

In the approach we take here — known as *option adjusted analysis* — it is assumed that the interest rate contingency is priced by the market, and hence the purpose of the analysis is to calculate the option adjusted spread (*oas* for short). This quantity indicates the incremental spread of the contingency over the Treasury yield curve, that is implied by the market price — a precise definition of *oas* is given in Section 2.3, equation (13). The significance of this measure in the context of investment decisions is documented in Hayre and Lauterbach [1988] for mortgage-backed securities, and in Asay et al. [1989] for insurance products.

The overall design of the valuation model is illustrated in Figure 1. We discuss next in detail each phase of the model, and continue in Section 3 with a discussion of the parallel implementation of each phase.

2.1 Phase I: Generating Scenarios of Interest Rate Paths

Models for the evolution of the term structure of interest rates are of central interest in financial modeling applications. Two commonly used procedures — for which we have developed massively parallel methods — are based on (1) the Monte Carlo simulation of a diffusion process, and (2) on the sampling of binomial lattices.

2.1.1 Simulation of a Diffussion Process

In this model interest rates are assumed to follow a lognormal distribution with "mean reversion" modifications that keep the interest rate paths within historically acceptable bounds. We use a model adopted in some Wall Street studies, like Hayre and Lauterbach [1988]. Drift factors are used to calibrate the model for consistency with the term structure of interest rates.

The one-year forward rate f_t at time $t \in \{1, 2, 3, \ldots, T\}$, is generated from the rate at period $t-1$ as follows:

$$\log \frac{f_t}{f_{t-1}} = z * \sigma_t + R(f_t) + \mu_t \qquad (1)$$

where:

z is a normally distributed random variable, $z \sim N(0,1)$,

σ_t is the volatility of interest rates per unit time, at instance t,

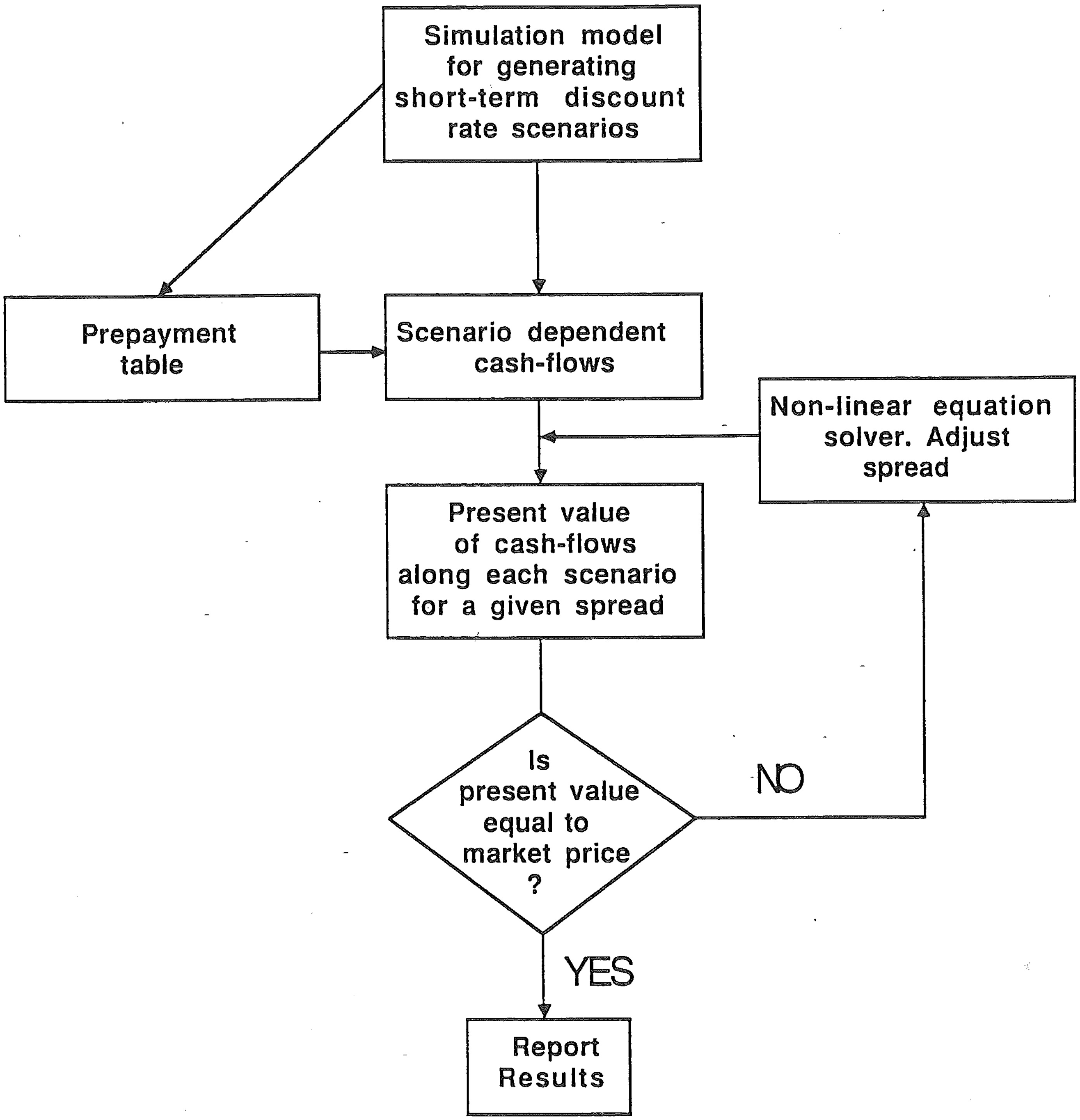

Simulation model for generating short-term discount rate scenarios
Prepayment table
Scenario dependent cash-flows
Non-linear equation solver. Adjust spread
Present value of cash-flows along each scenario for a given spread
Is present value equal to market price ?
NO
YES
Report Results

Figure 1: The Option Adjusted Analysis Model.

$R(f_t)$ is a mean-reversion term that forces the simulated rates to stay within historically acceptable limits defined by

$$R(f_t) = \begin{cases} 0 & if \;\; l \leq f_t \leq u \\ -\gamma_0(f_t - u)^2 & if \;\; f_t > u \\ \gamma_1[(l - f_t)/f_t]^2 & if \;\; f_t < l \end{cases} \tag{2}$$

where γ_0 and γ_1 are constants estimated from empirical data.

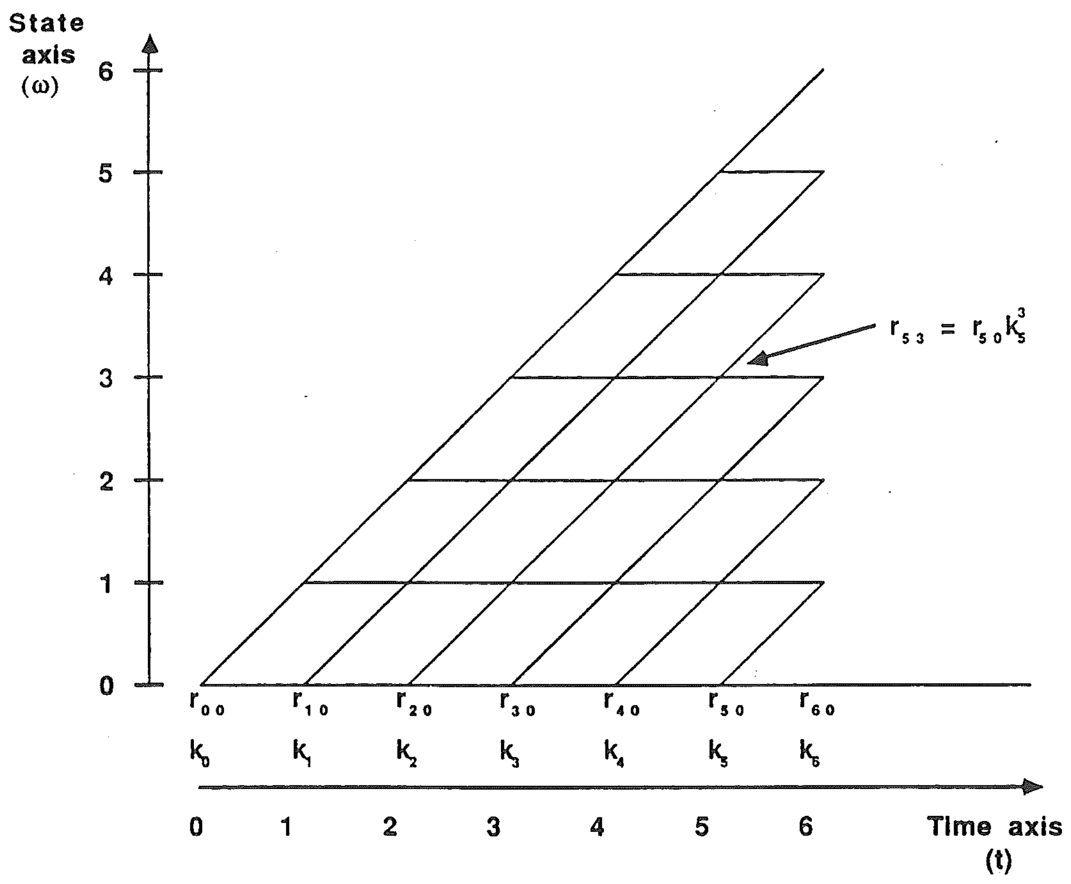
μ_t are drift factors estimated so that the model rates are consistent with the term structure. They are estimated by requiring that the present value of on-the-run treasuries, computed using the model rates, is in agreement with the current market prices.

A large number of interest rate paths can be generated by repeated application of equation (1).

2.1.2 Sampling from a Binomial Lattice

An alternative approach to the simulation of a diffussion process is to assume that term structure movements can be approximated by a discrete binomial process, represented by a lattice as shown in Figure 2. Discrete points in time are marked on the horizontal axis, and nodes of the lattice represent possible states of interest rates at every point in time. The term structure can move to one of two possible states between successive points in time — conveniently called the "up" and "down" states. The lattice is connected in the sense that an "up, down" and a "down, up" path starting from the same state will lead to the same state after two periods. After t time periods from the origin the lattice has t possible states. Each one of this states can be reached through 2^t possible paths. For example, a binomial lattice of interest rates over a 30 year time horizon in monthly intervals has 360 possible states at the end of the planning horizon, and a formidable 2^{360} possible paths of interest rates that lead to these states.

Short term forward rates at the nodes of the lattice can be computed based on market data, in such a way that the arbitrage free property is satisfied. Alternative techniques for fitting a binomial lattice to market data are beyond our present task. Readers are referred to Ho and Lee [1986] or Sharpe [1985] for comprehensive discussions, and to Bookstaber, Jacob and Langsam [1986] for critique. In our work we use the single factor model of Black, Derman and Toy [1987].



State axis (ω)
6
5
4
3
2
1
0
$r_{53} = r_{50}k_5^3$
r_{00} r_{10} r_{20} r_{30} r_{40} r_{50} r_{60}
k_0 k_1 k_2 k_3 k_4 k_5 k_6
0 1 2 3 4 5 6
Time axis (t)

Figure 2: Binomial Lattice of Interest Rates.

Once the binomial lattice has been fitted to the current term structure — in itself a difficult and compute intensive process — we can represent the short term rate at time period t and at state ω by the relation

$$r_{t\omega} = r_{t0}k_t^{\omega} \tag{3}$$

The quantities r_{t0}, k_t for $t = \{1, 2, 3, \ldots, T\}$ represent the 0-th (i.e., ground) state, and the volatility of short term rates at period t; they are parameters estimated by the binomial lattice model of choice. A large number of interest rate scenarios are computed by sampling paths from the binomial lattice using equation (3). A massively parallel procedure for this task is presented in Section 4.1.2.

2.2 Phase II: Cashflow Calculations for Mortgage-Backed Securities

We continue the presentation of the valuation framework by reviewing cashflow generation methods. For a comprehensive treatment of this topic refer to Fabozzi [1987] or Bartlett [1988].

The cashflow generated by a MBS at discrete points in time $t \in \{1, 2, 3, \ldots, T\}$, consists of three components:

1. *Interest payment:* is the portion of cashflow that reflects interest. It depends on the mortgage contract interest rate and the outstanding mortgage balance, net any servicing fee that is kept by the lending institution.

2. *Scheduled principal payment:* is the scheduled payment of outstanding principal.

3. *Projected principal prepayment:* represents any cashflow generated due to the exercise of the call option by some homeowners. For example, due to sale of the property or refinancing of the loan, some mortgages in the pool will pay a lump sum equal to the outstanding mortgage balance.

   The principal prepayment is estimated based on projected monthly *prepayment rates* that indicate the fraction of outstanding balance that is prepaid. The estimation of prepayment characteristics of a pool of mortgages is a cenral issue in mortgage backed financing. Severl models have been proposed and are used in practice with varying levels of success. A simple model, that is quite often used as benchmark inspite its simplicity — or, perhaps, due to it — is the PSA (i.e. Public Securities Association) model. Acording to the PSA model prepayment rates increase linearly for the first 30 months of the mortgage until they reach a rate of 6% for t=30, and then remain constant for the rest of the life of the mortgage. A discussion of the key prepayment factors that leads to a more acurate model can be found in Richard and Roll [1989]. A comprehensive prepayment model, developed jointly by researchers at Union Bank of Switzerland and one of us is reported in Hill, Pan and Zenios [1989].

   For the purpose of this study we obtained projected prepayment rates in the form of monthly survivorship factors — the complement of prepayment — generated by the prepayment model of Bear-Stearns. These data is provided in a tabular form for each MBS of interest. Each table has as its dimensions the range of possible interest rates — e.g., 6% to 16% — and the age of the security. If the one-year forward rate from the simulation model is f_t in month t, we simply look up the monthly survivorship in the tth column of the table, and at the row that corresponds to the interest rate closest to f_t. These prepayment tables are large datasets: The range of interest rates is divided into small increments (e.g., .01 basis points) for a total of 1000 increments in the range 6% to 16%. The maximum number of months for the security is 360, and each table has 360,000 entries.

Each one of the three cashflow components can be calculated based on the characteristics of the MBS and the prepayment rates. We define first the following:

n: Original outstanding term of the mortgage measured in months.

i: Simple monthly interest rate of the mortgage.

s: Servicing fee rate.

For each period $t \in \{1, 2, 3, \ldots, T\}$ we also define:

mb_t: Outstanding mortgage balance at the end of period t with mb_0 denoting the current outstanding mortgage balance.

mp_t: Scheduled mortgage payment without adding prepayment or netting the service fee.

I_t: Interest payment.

The cashflow calculations now proceed as follows; see Fabozzi [1988]. The scheduled monthly payment is

$$mp_t = mb_{t-1} \frac{i(1+i)^{n-t+1}}{(1+i)^{n-t+1} - 1} \tag{4}$$

and the scheduled interest payment is

$$I_t = mb_{t-1} \cdot i\,. \tag{5}$$

The net interest ni_t received by security holders in month t is this interest less the servicing fee, or

$$ni_t = mb_{t-1} \cdot (i - s) \tag{6}$$

The scheduled principal payment, sp_t, is the monthly payment less the mortgage interest, or

$$sp_t = mp_t - I_t \tag{7}$$

The amount of principal prepaid is determined by the monthly survivorship factor, x_t. The prepayment in month t of principal, pr_t, is calculated as follows:

$$pr_t = (1 - x_t)(mb_{t-1} - sp_t) \tag{8}$$

The net cashflow to investors in this pass-through security in month t, cf_t, is the sum of net interest, scheduled principal payments, and principal prepayments:

$$cf_t = ni_t + sp_t + pr_t \tag{9}$$

Finally, the mortgage balance is computed recursively by the equation

$$mb_t = mb_{t-1} - (sp_t + pr_t) \tag{10}$$

Example 1: We illustrate the cashflow calculations for a new MBS with original balance \$100,000, mortgage rate 9.5% and servicing fee 0.5%. We assume deterministic prepayment rates that increase according to 50% of the benchmark PSA model at $0.5\frac{6\%t}{30}$ for $t < 30$ and are constant at 3% thereafter. The monthly survivorship x_t is calculated according to

$$\begin{aligned} x_t &= \left(1 - 0.5\frac{6\%t}{30}\right)^{1/12} \quad \text{for } t < 30, \text{ and} && (11)\\ x_t &= (1 - 3\%)^{1/12} = 0.997465 \quad \text{for } t \geq 30. && (12)\end{aligned}$$

Table 1 summarizes the componenets of the cashflow calculation for selected months. In Figure 2 we illustrate the cashflow for this security assuming prepayment rates at 50% of the PSA model. Note that the PSA model assumes that the level of prepayment is independent of the prevailing interest rate, and is thus a crude approximation of reality.

For each iteration of the simulation, we determine the sequence of survivorship factors $\{x_t\}$ from the prepayment table, and then compute the cashflow by month for the security under analysis using equations (4) – (10). These cashflows are then used in the option adjusted analysis model described next.

t	x_t	mb_{t-1}	I_t	mp_t	sp_t	pr_t	cf_t
1	.999917	100,000	792	841	49	8	808
2	.999834	99,942	791	841	50	17	816
3	.999750	99,876	791	841	50	25	824
4	.999667	99,801	790	840	50	33	832
5	.999583	99,718	789	840	51	42	840
...							
29	.997551	95,158	753	813	59	233	1,007
30	.997465	94,866	751	811	60	240	1,013

Table 1: Example cashflow calculation of a new MBS with \$100,000 original balance, mortgage rate of 9.5%, servicing fee 0.5%. Prepayment rates are assumed to be equal to 50% of the PSA model.

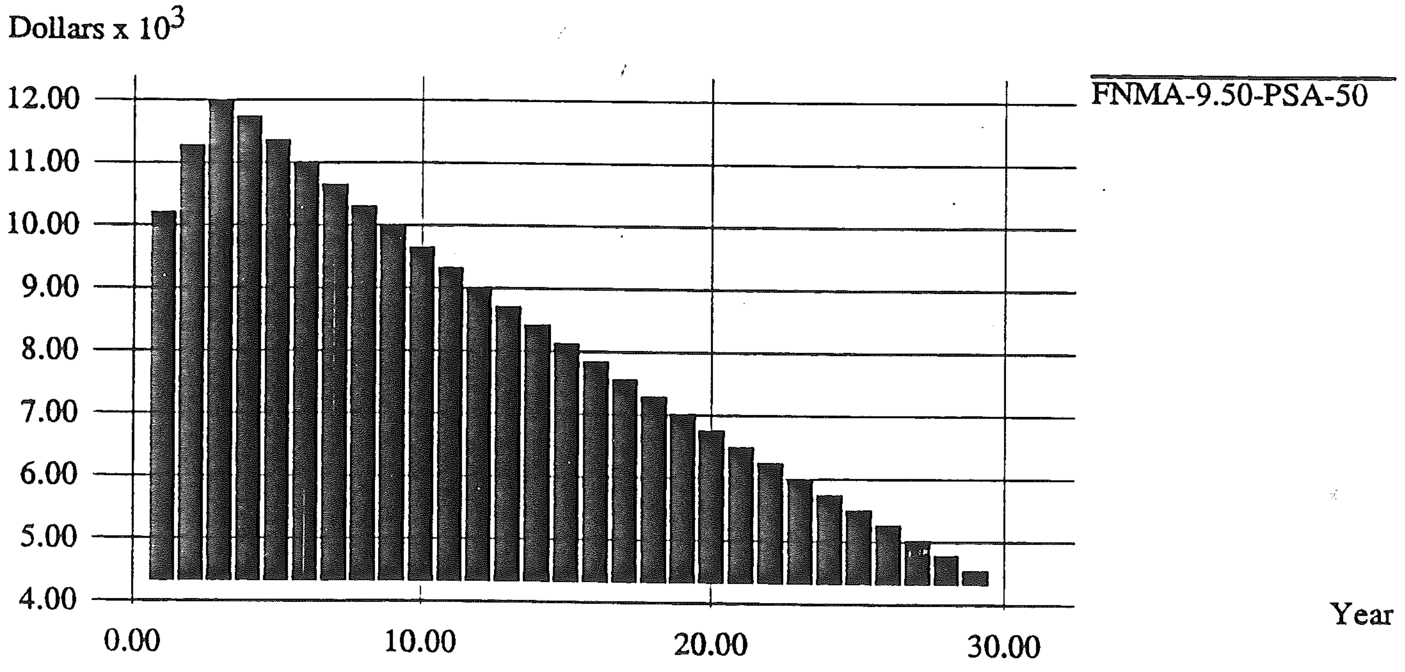


Figure 3: Cashflows generated by the model for a new MBS with \$100,000 original balance, mortgage rate of 9.5%, servicing fee 0.5%. Prepayment rates are assumed to be equal to 50% of the PSA model.

2.3 Phase III: Option Adjusted Analysis

For each iteration of the simulation the model generates a series of short term forward rates, estimated cashflows at every point in time, and the market price of the security. Let

S , be the sample of interest rate scenarios (paths) with cardinality $| S |$,

r_t^s , be the short term forward rate at time period $t \in \{1,2,3,\ldots,T\}$ under scenario $s \in S$, generated either by the Monte Carlo simulation or by sampling a binomial lattice.

cf_t^s , be the cashflow at time period t under scenario $s \in S$.

The option adjusted spread is the incremental spread over the short term rates that equates the expected present value of the cashflows under all scenarios with the market price. If P denotes the market price, the option adjusted spread is obtained by solving for oas the nonlinear equation

$$P = \frac{1}{| S |} \sum_{s \in S} \left\{ \sum_{t=1}^{T} cf_t^s \prod_{\tau=1}^{t} \frac{1}{(1 + r_\tau^s + oas)} \right\} \tag{13}$$

We assume in the sequel that a fair price for the security (P) is determined by the market, and we want to estimate the value of oas that will satisfy equation (13). Hence, we need to solve a nonlinear equation in the unknown oas. This nonlinear equation is solved using the algorithm of Van Wijngaadent-Dekker-Brent, see Press et al. [1988,Ch. 9.3], that generates a sequence $\{oas_k\}$ until the difference between the right and left hand sides of equation (13) is less than a user specified tolerance (10^{-4}). For every trial point oas_k of the nonlinear equation solver we have to evaluate the expression on the right hand side of equation (13). The massively parallel evaluation of this expression is the topic of Section 4.3.

3 The Connection Machine CM–2 System

3.1 Hardware Overview

The Connection Machine, see Hillis [1985], is a fine grain massively parallel supercomputer. It uses a single instruction stream, multiple data stream (SIMD) methodology: each processor executes the same instruction broadcast by a microcontroller on its unique piece of data, or optionally sits out of the computation. The CM–2 has from 4096 to 65536 bit-serial processing elements, each with local memory of either 8 or 32 Kbytes. Each processor also has access to 32-bit or 64-bit floating point acceleration hardware.

The interconnection scheme for processors in the CM–2 is an N-dimensional hypercube. The hardware supports two distinct types of communication patterns: local grid-based patterns, and general patterns. Local grid-based patterns are supported through direct use of the hypercube wires, while general patterns are supported through the *router*, which implements arbitrary point to point communication.

The CM–2 is controlled by a serial front-end computer. Programs are compiled, debugged, and executed on the front-end computer, passing CM–2 instructions to the microcontroller as appropriate. Data can also be passed either way along this path.

3.2 Programming Model Overview

The programming model for the Connection Machine is called data parallel computing, which means that the same computations are performed on many data elements simultaneously. Each data element is associated with a processor of its own. Applications are not restricted, however, to data sets matching the physical size of the machine. The Connection Machine system software supports the abstraction of an arbitrary number of *virtual processors* (VPs), allowing users to easily handle data sets with potentially millions of elements. VPs are implemented by partitioning the memory and time-sharing the cycles of each physical processor. A collection of virtual processors used to handle a data set is called a *VP set*, and the number of virtual processors that each physical processor must emulate is called the *VP ratio*. Note that because of pipelining and other optimizations, the expected linear slowdown from implementing VPs is actually a worst case scenario.

Applications that have no dependencies between data elements are often called *embarrassingly parallel*, and are easy to implement efficiently on any parallel computer. Most applications, however, exhibit dependencies between data elements. This gives rise to the need for communications between the processing elements. Furthermore, there is often a natural physical shape to arrange the data, such that dependent elements are laid out close to one another. Weather simulations, for instance, are naturally laid out in a three dimensional grid representing the volume of atmosphere being simulated.

The Connection Machine software allows users to specify the shape of a data set by associating a *geometry* with the VP set of the data. Geometries can be any N-dimensional Cartesian grid. Because an N-dimensional hypercube can be projected onto any lower dimensional Cartesian grid, the software can lay out the grid on the machine so that there is either a hypercube wire between each neighboring grid point, or they are held in the same physical processor (i.e., reside on VPs that are emulated by the same physical processor), and therefore local communication is fast and efficient. This kind of communication is called NEWS communication, and processors in the grid can be uniquely identified by the N-tuple of Cartesian coordinates called its NEWS address. General router-based communications is performed with a single unique identifier for each VP called its send address.

An important class of Connection Machine instructions that combine computations and communications are the parallel prefix operations, (Blelloch [1990]). We discuss the *scan* and the *spread* primitives here, although other variants exist. These primitives apply associative binary arithmetic operators to a variable in a binary combining tree along one axis of the geometry that holds the variable. For example, a plus-scan along the first axis of a 1-dimensional VP set for the variable x[0],x[1],...,x[n] produces the result y[0],y[1],...,y[n], where y[i]=x[0]+x[1]+...+x[i] (i.e., the cumulative sum to that point in the axis). User options allow the result to omit a processor's own value for x (e.g., y[i]=x[0]+x[1]+...+x[i-1]) and/or for the operation to proceed in the reverse order (e.g., y[i]=x[i]+x[i+1]+...+x[n]). A spread is a scan without directionality. The result of an add-spread is defined by y[i]=x[0]+x[1]+...+x[n].

The primitives have natural extensions to multi-dimensional geometries. For example, a plus-scan operation on the set of variables x[0][0], ... x[m][n] along the first axis of a 2-dimensional geometry, will produce the result y[i][j] = x[i][0]+x[i][1]+ ... + x[i][j], for all i=1,2,3,...,m, and j=1,2,3,...,n. Finally, note that with a binary combining tree implementation, the execution time of such operations is proportional to the logarithm of the number of VPs, and thus scales nicely to large data sets.

4 Massively Parallel Computing for Option Adjusted Analysis

The simulation procedure parallelizes nicely if each processor carries out all computations for a single interest rate path. Multiple processors can then execute in parallel multiple simulations. Communication across processors is only required in computing statistics across all simulations. It has been shown in Zenios [1990] that a parallel implementation along these lines can be very efficient on shared- or distributed-memory parallel architectures with a small number of processors.

On a system like the CM–2, however, we want to exploit the massive parallelism in performing the calculations for each path, in addition to simulating multiple paths simultaneously. Otherwise, a large number of processing elements will remain unused and the performance of the program will fall far short of the typical performance of the hardware. The path dependencies that are inherent in the OAA methodology appear, on first examination, to preclude the exploitation of parallelism within each path. However, we have found formulations of the method that do allow this form of parallelism. The following sections illustrate the efficient parallel implementation of the primary components of the OAA model.

The key to our implementation is the configuration of the CM–2 virtual processors into a 2-dimensional NEWS grid. One dimension of the grid is equal to the desired number of simulations, which for computational efficiency is taken to be a power of 2. The second dimension of the grid is equal to the number of time periods rounded up to the next integer that is a power of 2. The OAA model is usually developed on a 30 year time horizon in monthly intervals (i.e., T=360 time periods). Also, the number of simulations is taken to be 1024. With the use of variance reduction techniques these many simulations are sufficient to reduce the sampling error to an acceptable level. Hence, a NEWS grid of dimension 1024×512 suffices. Each one of the 1024 rows of virtual processors of the 0-axis carry out the calculations for a single path. The first 360 virtual processors in each row execute the path dependent calculations (the remaining 152 are idle).

Since the CM supports the notion of virtual processing it is possible to increase either the number of simulations or the number of time periods, or both. No modifications are needed in the algorithms, or in the code. On a CM–2a with 4K processing elements the 1024×512 NEWS grid requires a VP ratio of 128. The highest VP ratio at which the program could be executed on the same size CM with 32Kbytes of memory is 256. At this ratio we could implement a NEWS grid for 2048 simulations for more than 40 years in monthly intervals. Large systems can also be used.

In the sequel we give details of the implementation for each one of the three phases of the model. Virtual processors in the 0-axis of the grid are indexed by $s = 1, 2, 3, \ldots, 1024$ indicating simulation paths, and virtual processors in the 1-axis are indexed by $t = 1, 2, 3, \ldots, 360$ indicating time. Calculations along the 1-axis (time) are explained in detail, and it is to be understood the identical calculations are carried out simultaneously by all the VPs along the 0-axis (simulation).

4.1 Phase I: Parallel Generation of Interest Rate Scenarios

We have implemented scenario generation procedures based on both Monte Carlo simulation and the sampling of a binomial lattice.

4.1.1 Monte Carlo Simulation

The generation of mean-reverting lognormally distributed series is not an associative operation, due to the mean-reversion term (2). Hence, an implementation that uses the scan primitives efficiently is not possible. It is possible, for example, to generate the lognormally distributed series first using the scans, and then proceed to apply the mean-reverting equation. However, this approach implies that whenever $R(f_t) \neq 0$ (see equation (2)) the generated series for periods t to 360 has to be discarded, the mean-reversion term $R(f_t)$ added, and a lognormally distributed series generated anew using the scan primitives. This approach may lead to as many as 360 repeated uses of the scan primitives and was observed, in a preliminary implementation, to be inefficient.

The Monte Carlo simulation phase was implemented by marching forward along the time-axis in steps of 1 and generating in parallel the states of 1024 interest rate paths at each point in time. Of course this implementation requires the use of only 1024 virtual processors. Hence, instead of operating on the NEWS grid of dimension 1024 $\times$ 512 we implement this primitive on a NEWS grid of dimension 1024 $\times$ 4 at a VP ratio of 1. Now it is possible to associate each column of VPs of the 1024 $\times$ 4 grid with a column of the 1024 $\times$ 512 grid. Doing so we facilitate the transfer of data from the 1024 $\times$ 4 grid where they are generated to the 1024 $\times$ 512 grid where they are used for subsequent analysis without the need to use router communications.

As will be shown in the section on computational results the generation of mean-reverting interest rate paths is the most expensive part of the simulation. Futhermore, its performance does not scale with the availability of larger number of processors since it is already executed at a VP ratio of 1.

4.1.2 Sampling from a Binomial Lattice using Scan Operators

In order to generate sample paths from the binomial lattice we need to determine the state of each path at each point in time. Once the state ω of the sth path in the binomial lattice is specified at virtual processor with NEWS address (s, t) the short term rate can be computed by a simple application of equation (3). The problem in constructing a continuous path is that the state of the lattice at instance t must be attainable by either an "up" or "down" step from the state at instance $t-1$. Such a sequence of states is produced on the CM–2 as follows: A random bit $m_t \in \{0,1\}$ is first generated at each VP. A *scan-add* operation along the time axis on these bits generates an index (ω_t), indicating the state of the VP (i.e., its distance from the ground state r_{t0}). Clearly, the distance from the ground state at instance t differs by at most one unit from the distance at instance $t-1$. Once the distance ω_t is determined equation (3) can be evaluated simultaneously by all VPs. Figure 3 illustrates the sampling of two paths from a binomial lattice using this procedure.

An alternative way to the sampling of binomial lattices has been developed by Shtilman and Zenios [1990]. Their method eliminates the need to construct random path samples from the lattice. Instead, it generates a prespecified set of paths. These paths will provide estimates for the path dependent discount functions that are within a user specified error from the exact value, with a given probability. These paths are constructed by sampling <u>all</u> possible paths from the first $\mathcal{T}$ time instances of the lattice and completing each path from $\mathcal{T}+1$ to T by any arbitrary sequence of change of state.

This sampling scheme is implemented as follows. The CM–2 is configured as a two-dimensional NEWS grid of dimensions $2^{\mathcal{T}} \times 512$. The binary representation of row index s of this grid specifies the change of states (i.e., m_t) for the sth path from instance 0 to $\mathcal{T}$. The change of states from instance $\mathcal{T}+1$ to 360 is completed by the (arbitrary) sequence $\{01010101...\}$. Once the binary sequence is completed we use once more a *scan-add* operation to evaluate the state ω_t of each VP and equation (3) is used to evaluate the short term rates. Figure 4 illustrates the sampling of all paths from the first three time periods of a binomial lattice.

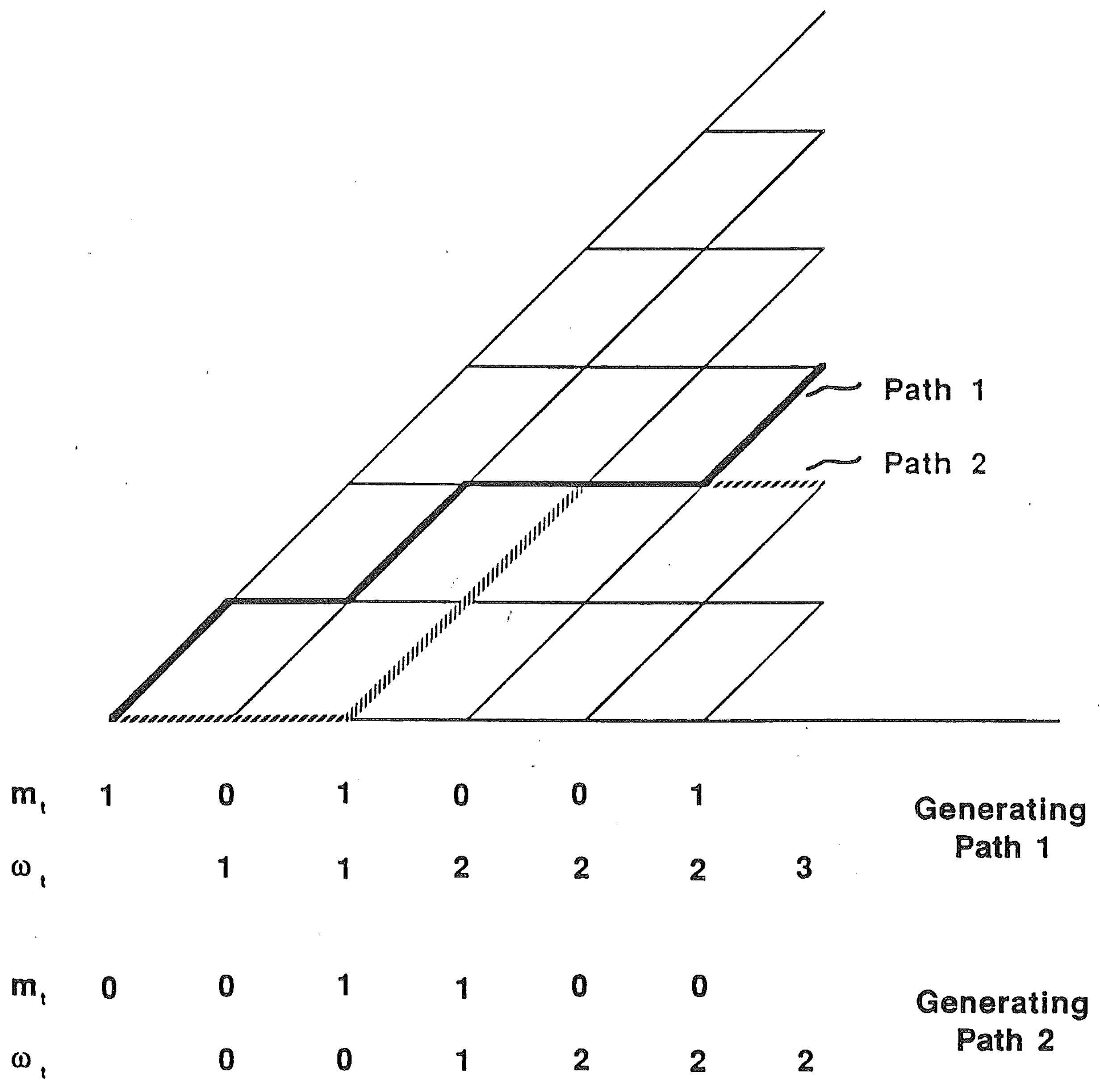
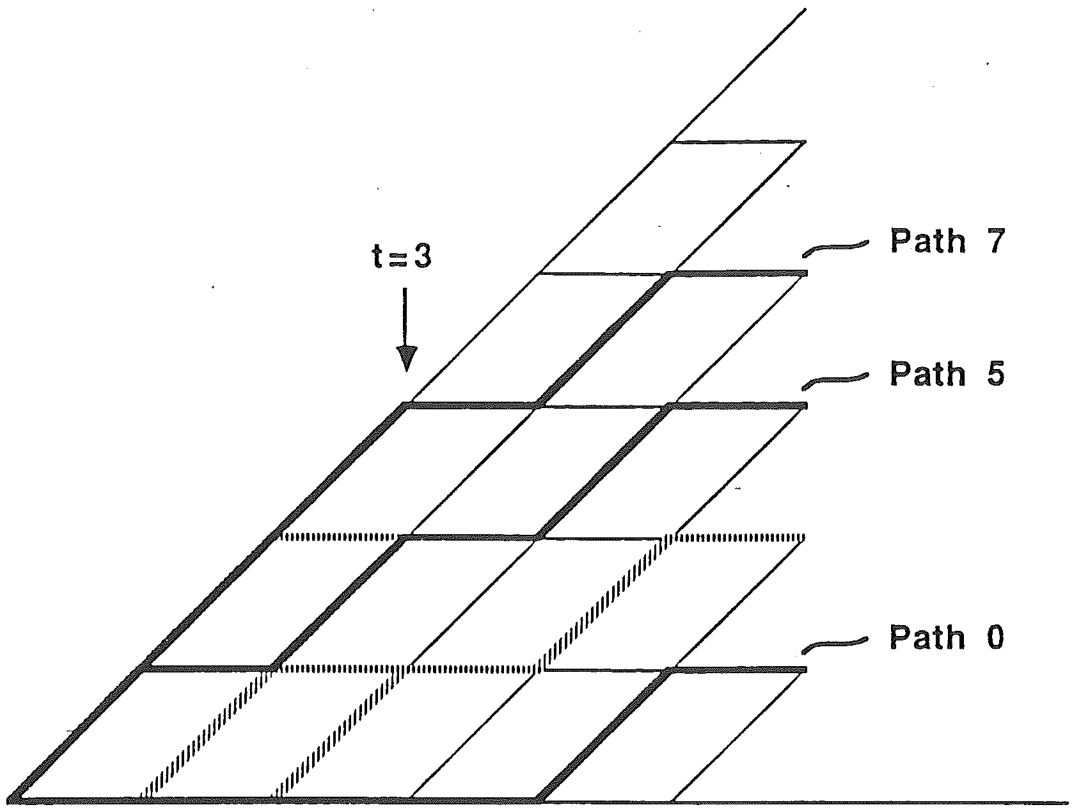


Figure 4: Sampling paths from a binomial lattice on the CM-2. $m_t = \{0, 1\}$ indicates a transition to the "down" or "up" states at instance t. ω_t indicates the distance from the ground state r_{t0}



Path 0: **Binary representation** **0 0 0**

m_t = 0 0 0 0 1 0

Path 5: **Binary representation** **1 0 1**

m_t = 1 0 1 0 1 0

Path 7: **Binary representation** **1 1 1**

m_t = 1 1 1 0 1 0

Figure 5: Sampling all paths from the first three time periods of a binomial lattice on the CM–2. $m_t = \{0, 1\}$ indicates a transition to the "down" or "up" states at instance t. ω_t indicates the distance from the ground state r_{t0}

4.2 Phase II: Parallel Cashflow Calculations

There are two aspects of the cashflow model that deserve some attention for parallel implementation: a table lookup function to get the prepayment rates from the database, and the calculation of all the components of the cashflows. The cashflow calculation appears particularly challenging due to the recursive nature of the mortgage balance equation (10).

4.2.1 Lookup of Prepayment Tables

We consider here the case where the prepayment characteristics of a mortgage-backed security have been generated by a separate model and are stored in a table of *monthly survivorship factors.* This is not an unreasonable modeling practice: prepayment characteristics of mortgage securities are re-evaluated rather infrequently, whereas the OAA model is usually run on a daily basis, or, potentially, in real time. [5]

Our problem is then one of looking up the correct monthly survivorship factor for the given level of interest rates and age of the mortgage. The monthly survivorship data is given in a two-dimensional table. The 1-axis indicates the age of the mortgage (1 – 360 months) and the 0-axis indicates the interest rate in a given range (e.g., 6 – 16%) in small increments (e.g., 0.01). This table is stored in a two-dimensional NEWS grid. The dimension of the 1-axis is 512, with only the first 360 columns being active. The dimension of the 0-axis depends on the range and increment of interest rates. In the example given above the 0-axis has dimension 1024 with only the first 1000 rows being active. It is only by coincidence that the dimension of the 0-axis of the NEWS grid used for the simulation is identical to the size of the grid required by the survivorship table. Hence, these two grids are associated with different VP sets. We shall refer to these sets as the *simulation-vp-set* and *data-vp-set* respectively.

Looking up the survivorship factors given a path of interest rates is now a simple data transfer from the data-vp-set to the simulation-vp-set. Virtual processor with coordinates (s, t) in the simulation-vp-set will find the row index i of the interval in the data-vp-set that corresponds to interest rate r_{st}. Let R_0 be the lowest interest rate in the survivorship table, and δ be

[5] The massively parallel implementation of prepayment models is the topic of current research and will be reported elsewhere.

the increment between successive rows. Then the row index is computed by

$$i = \lfloor \frac{r_{ts} - R_0}{\delta} \rfloor \tag{14}$$

The (s,t)*th* VP from the simulation-vp-set will get from the (i,t)*th* VP of the data-vp-set the survivorship factor x_t using router communications. This factor is then used in the cashflow calculations.

4.2.2 Cashflow Calculations using Scan Operators

The parallel evaluation of the cashflow equations of Section 2.3 presents some difficulties due to the recursive equation (10). It is possible to develop an alternative formulation of the cashflow equations that lends itself to the application of the scan parallel primitives.

Let

$$c_t = \frac{i(1+i)^{n-t+1}}{(1+i)^{n-t+1} - 1} \tag{15}$$

and

$$\rho_t = (1 - x_t)(1 - c_t + i) \tag{16}$$

After some algebra using equations (4) — (9) we may rewrite the recursive mortgage balance equation (10) as

$$mb_t = mb_{t-1} \cdot \rho_t \tag{17}$$

This in turn can be rewritten as

$$mb_t = mb_0 \prod_{\tau=1}^{t} \rho_\tau \tag{18}$$

Equation (18) can now be evaluated using a *scan-multiply* operation that will produce at the t*th* VP in each row of the NEWS grid the value $\prod_{\tau=1}^{t} \rho_\tau$, and a multiplication of the result by the constant mb_0 that will produce the correct mortgage balance exclusive of prepayments. Scheduled payment, net interest, prepayment and total cashflow at each point in time can then be evaluated simultaneously for all $t = 1, 2, 3, \ldots, T$ by the sequence of equations

$$sp_t = mb_{t-1}(c_t - i) \tag{19}$$

$$ni_t = mb_{t-1}(i - s) \tag{20}$$

$$pr_t = (1 - x_t)(mb_{t-1} - sp_t) \tag{21}$$

$$cf_t^s = sp_t + ni_t + pr_t \tag{22}$$

Phase III of the model uses the cashflow cf_t^s at each period and for each scenario to calculate the present value, or compute the option adjusted spread.

4.3 Phase III: Present Value Calculation using Scan Operator

The present value calculation that appears in equation (13) can be written in the form:

$$PV = \sum_{t=1}^{T} cf_t^s \prod_{\tau=1}^{t} \frac{1}{(1 + r_\tau^s + oas)} \tag{23}$$

A *scan-multiply* operation on the ratio $\frac{1}{(1+r_\tau^s+oas)}$ produces at virtual processor with NEWS address (s,t) the value $\rho_t^s = \prod_{\tau=1}^{t} \frac{1}{(1+r_\tau^s+oas)}$. Each virtual processor proceeds to multiply the local values of cf_s^t with ρ_t^s and a *scan-add* on the product completes the calculation.

5 Computational Results

The procedures of the previous section were implemented in C/Paris for the Connection Machine as part of a library of financial routines, and this library was used to build an OAA model. The library was built on top of Paris release 5.2. The results we report here are obtained using single precision (i.e., 23 bits in the mantissa) arithmetic, although double precision (i.e., 52 bits in the mantissa) is also available. A partial listing of the entries of the library is given in the Appendix.

Individual modules of the library were first tested both for efficiency in terms of solution times, and in terms of the sustained computing rate. All experiments were carried out on a 4K processor model CM–2a at Thinking Machines Corporation, with a Sun-4/280 front-end, 32-bit floating point accelerators and 8 Kbytes of memory per processor. In our implementation we use a 1024×512 NEWS grid for a total of 2^{19} virtual processors. Hence, the implementation is executed at a VP ratio of 128, with the exception of the Monte Carlo simulation that is implemented at a VP ratio of 1 on a 1024×4 grid. Some improvements in performance will be realized for all functions, except the simulation of the diffussion process, if the programs were run on a bigger machine. Increasing the number of processing elements can not reduce the VP ratio for the simulation of the diffussion process, and

Function	VP ratio	CM time (seconds)	Elapsed time (seconds)	MFLOPS
Present value	128	0.15	0.15	93
Monte Carlo simulation	1	0.50	0.53	58
Random sampling of binomial lattice	128	0.24	0.25	59
Prepayment table lookup	128	0.27	0.28	NA
Cash-flow calculations	128	0.23	0.23	117

Table 2: Execution times (seconds) for 1024 simulations on the Connection Machine CM–2a with 4K processing elements.

will not improve the execution time. All other functions of the OAA model are executed at a very high VP ratio. Doubling the number of processing elements on the CM will reduce by a factor of 2 the VP ratio. The execution time will reduce by a factor which is close to 2. (The reason solution time does not decrease linearly with the increase in the number of processing elements has to do with the way the VP system is implemented on the physical hardware: For higher VP ratios more communication is localized on a physical sprint node. As the VP ratio is reduced more communication has to take place accross the hypercube network.) Table 2 summarizes the execution time for the key components of the library in executing a total of 1024 simulations, together with the sustained MFLOPS rate. The MFLOPS rate is estimated by adding the number of FLOPS executed by each Paris instruction [6] , and then multiplying by the number of active processing elements, and dividing by the execution time. These rates are included in order to indicate how well our implementations are suited to the CM architecture. The cashflow calculations achieve very high computing rates, since several floating point operations are carried out (i.e., equations (16) and (19)–(22)) for each communication step (i.e., equaiton (18)). On the other hand, the sampling of a binomial lattice requires unstructured communications. The observed MFLOPS rate is substantial, but far from the peak performance of the CM. It is also interesting that the simulation of a diffussion process achieves substantial MFLOPS rate, even if it executes at a VP ratio=1. The reason is that there are no communications among processors during the simulations.

[6] Number of FLOPS for Paris instructions: 1 for addition and multiplication, 7 for division, 19 for ln, 16 for exp, 21 for *cos*, 17 for *sqrt*.

Mode	4K CM–2	8K CM–2	16K CM–2	CRAY X-MP
Monte Carlo Simulation	1.77	1.07	0.78	12.0
Binomial Lattice Sampling	1.49	0.71	0.37	6.0

Table 3: Performance of the option adjusted analysis model on the CM–2 and CRAY X-MP (seconds).

A complete option adjusted analysis was carried out for a single mortgage-backed security (GNMA) and was compared to an identical model implemented in Fortran 77 on a CRAY X-MP/48 vector supercomputer at CRAY Research Inc., Zenios [1990]. The solution time for both the CM–2 implementation and the CRAY X-MP code are reported in Table 3. Times are in CPU seconds on the CRAY and wall clock time on a dedicated CM.

The CRAY code was compiled using the f77 vectorizing compiler and was executed on a single CPU. The program was also executed in sinlge precision, which on the CRAY corresponds to 64-bit floating point arithmetic. The OAA model does not need more than 32-bit arithmetic and the results on the CRAY and the CM were almost identical. The average number of steps in the nonlinear equation solver reported by the Fortran program was 5, which again is consistent with the 4-6 iterations taken by the nonlinear equation solver on the CM.

It is anticipated that gains in performance can be realized with the CRAY code, with suitable modifications of the implementation to take advantage of the vector architecture beyond what is already achieved by the compiler. Multitasking the Fortran code to execute on 4 CPUs on the CRAY X-MP/48 would result in improvements of execution time by a factor of approximately 3.5. We expect that a fully vectorized and multitasked Fortran program on the CRAY would execute 7–10 times faster than the current implementation [7]. This would make the solution times of the X-MP/48 comparable to the 4K CM–2a. It would be interesting to see how a program written in Fortran 8X would execute on both systems without any modifications for vectorization, multitasking or the use of C/Paris instructions.

Finally, we provide in Table 4 benchmark results using the Fortran program on several systems, and the C/Paris program using Connection Machines of different sizes. At the same table we provide an estimate of the time needed to analyze a portfolio of 3000 MBS. The time needed for a

[7] This estimate is based on the results with the vectorization and parallel execution of the Fortran code on an Alliant FX/4; Zenios [1990].

complete portfolio analysis is estimated by multiplying the Monte Carlo simulation model time by 3000. This is an overestimation, since the interest rate scenarios will be generated only once and will then be used to calculate OAS for all securities in the portfolio. On the workstations, the VAX, and the CRAY, the scenario generation time is a very small fraction of the total simulation time. On the CM, however, the scenario generation is executed using a VP ratio=1 and represents a significant portion of the computation. For example, on the 16K CM–2 the scenario generation takes .5 seconds and the rest of the analysis only .28 (cf. Tables 2 and 3). If we take into account that only the .28 sec will be repeated 3000 times the total time for valuation of the portfolio on the 16K CM–2 is reduced from 40 to 15 minutes. On the fully configured 64K CM–2 the portfolio analysis can be completed in approximately 5 min. Similar line of reasoning leads us to an estimated time on the CRAY of about 8 hours on a single CPU. With vectorization and multitasking of the code on the X-MP/48 we estimate a total execution time of aproximately 1 hour.

A word of caution is due here: The estimates on potential improvements in performance is just that: estimates based on our prior experiences with both the CM and the CRAY, and numerical experiments carried out on similar architectures. They are not to be viewed as hard estimates, and the intention is to provide some insight on additional improvements that could be realized if the models were further fine-tuned for both the massively parallel system and the vector supercomputer. Nevertheless, the results reported in Table 4 are precise: they were obtained under identical conditions and levels of accuracy, and the answers of the model do not depend on the hardware platform.

6 Conclusions

Compute intensive financial simulations are well suited for execution on massively parallel architectures. While the execution of multiple simulations is easily parallelizable, the efficient execution of path dependent calculations is possible only with suitable modifications of the algorithms. It is then possible to implement the path dependent calculations by combining efficiently communications with computations via the parallel prefix primitives.

The performance of the massively parallel option adjusted analysis model on one of the smaller Connection Machine systems compares favorably with a CRAY X-MP vector supercomputer. As a result of very short response times

Computer	Monte Carlo simulation	Binomial Lattice sampling	OAS for 3000 MBS with Monte Carlo
Apollo DN4000	480	440	16 days
VAXstation	330	320	11 days
DECstation 2100	120	130	4 days
DECstation 3100	90	100	3 days
VAX 6400	60	60	2 days
CRAY X-MP (1 CPU)	12	6	10 hours
CM–2a (8K PE)	1.1	0.7	1 hour
CM–2a (16K PE)	.8	0.4	40 min.
CM–2a (32K PE)*	.7	0.2	35 min.
CM–2a (64K PE)*	.6	0.1	30 min.

Table 4: Benchmark Results with the OAS Model (CPU seconds). * Indicates estimated times, based on projections from the measured times with the 4K, 8K and 16K CM–2.

it is possible to use such models in real time and to carry out a wide range of sensitivity analyses. We have implemented an interactive user interface that allows users to compute option adjusted spreads, price, duration and convexity, projected price paths or average cash-flow statistics. The response time, including initializations, calculations and graph generations are only a few seconds (2-10) for most functions. Even the generation of a complete path of projected prices is generated well within half minute of wall clock time.

In Figure 6 we show the duration of a MBS under different prapayment rate assumptions as a function of interest rates. Duration is the derivative of price with small, local changes in interest rates. It is computed by carrying out a finite-difference approximation on the simulation program. This figure was generated in approximately one minute, including initializations and data display.

Massively parallel computer systems provided the computational tool for significant advances in several areas of science and engineering. Financial modeling applications stand to gain significantly from the use of this technology. It appears possible to employ sophisticated analytics for real-time trading applications. Even more importantly, however, it is now possible to build mathematical optimization models to do planning under the uncertain interest rate scenarios.

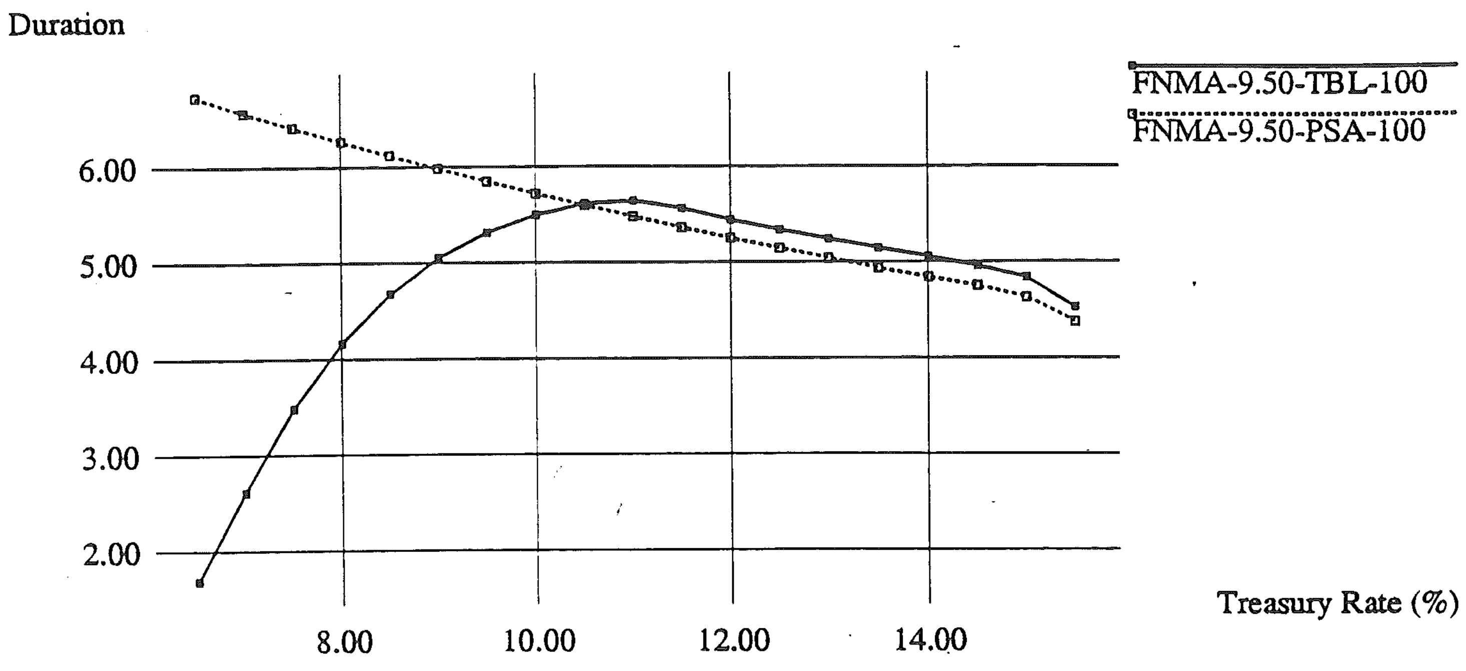


Figure 6: Effective duration of a mortgage-backed security assuming prepayment rates at (1) 100% of the PSA model, and (2) the rate given by the prepayment model. Each point on the curve is obtained using a finite difference approximation of the duration, from the prices computed by the simulation model.

A The Connection Machine Financial Library

We provide in the following table a list of the financial modeling primitives on the Connection Machine. Reported times are for the execution of calculations on a 1024 $\times$ 360 NEWS grid, on a 4K CM–2a with 32-bit floating point co-processors and a Sun4-280 front-end.

No.	Function	Real Time (seconds)	CM Time (seconds)	MFLOPS
1	prval	0.15	0.15	93
2	psa-prepayment	0.06	0.06	31
3	table-lookup	0.28	0.27	NA
4	rand-normal	0.25	0.25	94
5	rand-lognormal	0.42	0.40	73
6	rand-meanrev	0.53	0.50	58
7	sample-binomial-lattice	0.25	0.24	59
8	mortgage-cashflow	0.23	0.23	117
9	vector-mean	.00054	.00041	9

Acknowledgements: The work of S.A. Zenios was funded in part by NSF grant CCR–8811135 and AFOSR grant 89–0145. This research was completed while this author was visiting the Operations Research Center at the Sloan School of MIT, and Thinking Machines Corporation, Cambridge, MA. The support and useful discussions with J. Mesirov are greatfully acknowledged, as well as programming assistance and suggestions by A. Singer. Computing resources were made available by the North-east Parallel Architectures Center, Syracuse University, NY. The manuscript has benefited substantially from the comments of three anonymous referees.

References

[1] M.R. Asay, P.J. Bouyoucos, and A.M. Marciano. An economic approach to valuation of single premium deferred annuities. In *Financial Optimization*, Cambridge University Press. (to appear).

[2] W.W. Bartlett. *Mortgage-Backed Securities: products, analysis and trading.* New York Institute of Finance, 1988.

[3] F. Black, E. Derman, and W. Toy. *A One-factor Model of Interest Rates and its Application to Treasury Bond Options.* Discussion paper No. 1, Goldman Sachs, June 1988.

[4] G.E. Blelloch. *Vector Models for Data-Parallel Computing.* The MIT Press, Cambridge, Massachusetts, 1990.

[5] R. Bookstaber, D.P. Jacob, and J.A. Langsam. *Pitfalls in Debt Option Models.* Working paper, Morgan Stanley, Fixed Income Analytical Research, Feb. 1986.

[6] P.P. Boyle. Valuing canadian mortgage-backed securities. *Financial Analysts Journal*, 55–59, May-June 1989.

[7] A.J. Brazil. Citicorp's mortgage valuation model: option-adjusted spreads and option-based duration. *Journal of Real Estate Finance and Economics*, 1:151–162, 1988.

[8] F.J. Fabozzi, editor. *Mortgage Backed Securities: New Strategies, Applications and Research.* Probus Publishing Company, 1987.

[9] W. Foote, J. Kraemer, and G. Foster. APL2 implementation of numerical asset pricing models. *APL Quote Quad*, 1988.

[10] L. Hayer and K. Lauterbach. *Stochastic Valuation of Debt Securities.* Working paper, Financial Strategies Group, Prudential-Bache Capital Funding, 1988.

[11] H. Hill, P. Kang, and S.A. Zenios. *Complete prepayment models for mortgage-backed securities.* Working paper, Decision Sciences Department, The Wharton School, University of Pennsylvania, Philadelphia, 1990.

[12] W. D. Hillis. *The Connection Machine.* The MIT Press, Cambridge, Massachusetts, 1985.

[13] T. S.Y. Ho and S-B. Lee. Term structure movements and pricing interest rate contingent claims. *The Journal of Finance*, XLI(5):1011–1029, 1986.

[14] D.P. Jacob, G. Lord, and J.A. Tilley. A generalized framework for pricing contingent cash flows. *Financial Management*, August 1987.

[15] S.M. Pinkus, S.M. Hunter, and R. Roll. *An introduction to the mortgage market and mortgage analysis.* Mortgage Securities Research Series, Goldman Sachs, NY, 1987.

[16] R.B. Platt, editor. *Controlling Interest Rate Risk: New Techniques & Applications for Money Management.* John Wiley & Sons, 1986.

[17] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, Cambridge, 1988.

[18] S.F. Richard and R. Roll. Prepayments on fixed-rate mortgage-backed securities. *The Journal of Portfolio Management*, 73–82, Spring 1989.

[19] W.F. Sharpe. *Investments.* Prentice Hall, 1985.

[20] M. S. Shtilman and S.A. Zenios. *Constructing Optimal Samples from a Binomial Lattice.* Working paper, Decision Sciences Department, The Wharton School, University of Pennsylvania, Philadelphia, April 1990.

[21] N.E. Wallace and A. Wang. *The pricing of adjustable rate mortgages with a no-arbitrage binomial pricing model.* Technical Report, University of California, Berkeley, Dec. 1989.

[22] S. A. Zenios. Parallel monte carlo simulation of mortgage backed securities. In *Financial Optimization*, Cambridge University Press. (to appear).